Journal of the Association for Information Systems
JAIS

Special Issue

Balance of Trade in the Marketplace of Ideas

John Leslie King
University of Michigan
jlking@umich.edu

Abstract

If the Information Systems (IS) field is to exist with other fields in some kind of balance of trade in a marketplace of ideas, the scheme is not working too well, at least when comparing IS with Computer Science (CS). The trade tends to be one-way, from CS to IS. This paper explores why that is the case, and what might be done to change things.

***Keywords**: Information Systems, Computer Science, Marketplace of Ideas, Balance of Trade.*

* Nicholas Berente was the accepting senior editor. This article was submitted on 16th September 2012 and went through no revisions.

Volume 14, Special Issue, pp. 192-197, April 2013

1. Imbalance of Trade

If the information systems (IS) field is a marketplace of ideas (Lyytinen & King, 2004), one would hope for a balance of trade across the border between information systems and other fields, including computer science (CS). I do not find this to be the case. The IS field takes a lot from CS, but it does not send much back the other way. Why not? And what might be done about this?

The IS field draws a great deal from CS, starting with the artifacts of information technology. Orlikowski and Iacono (2001) argued that the IS field is (or should be) deeply linked to IT artifacts, and almost everyone in IS would say that digital computers are central to the IS field. Yet few people in CS would say that "management" or "organizations" or even "information" is central to what they do. CS has historically been reluctant even to admit "applications" to its list of key interests. When I joined the faculty of information and computer science at the University of California at Irvine in 1980, I became part of "Area 5". I later learned that this meant software and applications, but no one wanted to use those words openly. Applications are what make computing important, and CS students today demand attention to applications, but such changes have come slowly. IS looks toward CS for inspiration; CS seldom looks toward IS at all. In fact, CS sometimes fails to look to its own history for guidance (King, 2010).

This is not the only evidence that CS and IS have less than balanced trade in ideas. Who moves where is another indicator. There is relatively little faculty movement across the CS/IS border but, when it happens, it is more common for CS people to join IS programs than the other way around. This restricts knowledge transfer from IS to CS. IS people moving to CS often end up in databases, information retrieval, artificial intelligence, and computer-human interaction rather than in the "central" areas of CS, Theory and Systems (capitalized to indicate that these serve as names or signifiers rather than descriptions of the work being done). Exceptions can be found in Europe and Asia, but this generalization holds up pretty well in North America.

What explains the imbalance? Compensation might be a factor: on average, assistant professors in business administration (where most IS faculty are) earn 10 percent above what assistant professors in CS earn. Business school staff often receive summer salary and additional pay for executive education directly from their schools, while staff in engineering must often compete for sponsored research support from companies or federal research agencies. On strictly economic grounds, one would expect more CS PhDs to pursue employment in business schools. Yet they do not, in part, I argue, because the IS field holds little inspiration for them, or is ignored by the people they find most influential.

Another factor is culture in recruiting, which is grounded in the shared beliefs of individuals in academic fields. CS people have a stronger sense of the "core" of their field than do IS people, and have been less likely to look seriously at faculty candidates who are not steeped in that core. CS embodies an engineering worldview based largely in the physical sciences. If someone asks you the essence of CS, the reply "Theory and Systems" will probably satisfy them that you understand the field. In contrast, some in IS eschew the very idea of a "core" to the field (King & Lyytinen, 2006), and management schools are grounded in the social sciences, which are so broad that a worldview like that of engineering is unlikely.

Beyond compensation and culture in recruiting, there is the question of incumbency. The CS field is relatively new as academic fields go, but it is older than the IS field by at least a half-generation and possibly a full generation. One could argue that the two fields are separated by more than four decades (the Association for Computing Machinery, ACM, was founded in 1947, while the Association for Information Systems, AIS, was founded in 1994). A more realistic guess at age separation is about 15 years, or about a half-generation. The first PhDs in CS were given in the early 1960s, while the first PhDs in IS were given in the mid-1970s. The application of general-purpose digital computers to business in the early 1950s quickly followed the development of such machines in the late 1940s. Hiring was almost always at the bachelors or masters level because there were few PhDs in CS or IS to hire. Commercial organizations that made computers hired engineers and computer scientists, while companies that used computers hired both CS and IS people. University-based programs responded accordingly at the bachelors and masters levels. At least, from what I saw in the CS community, there was little pressure on CS PhD students to consider IS perspectives. Not surprisingly, CS faculty seldom have much perspective on IS.

2. A Broader View

All that said, CS has been evolving toward greater willingness to engage in cross-border exchanges in the marketplace of ideas. As CS grew beyond its mathematical roots, CS programs founded in mathematics departments needed equipment. Mathematics departments historically did not ask for equipment. Many CS programs left mathematics and migrated to engineering where equipment was easier to come by. CS programs joining engineering were often "anded" rather than "ored" because engineering itself was becoming more computational. Many CS groups joined electrical engineering to create electrical engineering and computer science, an arrangement still seen today. In contrast, IS programs in business schools co-resident with operations management or accounting have often been "ored" rather than "anded", which has created a form of "parallel play" wherein each group pursues its own interests. Both IS groups and CS groups have sought independence, but the CS side has been more successful of late than the IS side. Both CS and IS have, in some cases, created independent departments, but some CS programs have created independent schools focused on computing, IT, and information. IS people have crossed borders into CS or i-Schools, but few IS groups have established themselves as schools. CS people have been willing to look toward other fields, sometimes to fields outside of engineering. However, they seldom look to IS, despite the fact that important work on the transformation of the world with IT comes from IS researchers (e.g., Boland et al., 2001; Sawyer & Winter, 2011; Winter & Butler, 2011; Yates, 2005).

CS has been moving toward a broader worldview for some time, albeit not without controversy. This has been driven by the rising importance of computing in nearly everything, a force that should be pushing IS and CS closer together. There is some broadening of the IS field, but it has not been as extensive or contested as the CS field's efforts in this vein. Twenty years ago, the Computer Science and Telecommunications Board of the National Academy of Sciences published *Computing the future: A broader agenda for computer science and engineering* (Hartmanis & Lin, 1992). That report recommended that CS embrace what the report called "commercial computing", which includes model management, decision support, easy-to-use software, software development metrics and modeling, and collaborative work support. The report was controversial, and some distinguished computer scientists, arguing that attention to applications would weaken the field's focus on basic research, demanded that it be withdrawn (Bass, 1992).

The 1992 report was not withdrawn. Within fifteen years it seemed prescient. In 2007, the National Science Foundation created the Computing Community Consortium to explore the edges of computing research in areas as diverse as technology in K-12 education and IT in economic development. Members of the CCC Council, many of whom are leaders at the center of the CS field, were instrumental in preparing the influential PCAST-NITRD 2010 report on federal research funding in networking. That report touched on many issues of interest to the information systems community. The world demands these changes, and they are happening in spite of academic politics.

Can the IS community influence CS? This question should not be dismissed out of hand; it is possible that the IS community simply has little to offer CS. Perhaps CS people concern themselves with issues that few IS people care to deal with. To the extent that both CS and IS stay focused on issues they have historically cared about, I would not expect much movement from CS towards IS. If that is the case, the IS field would be wise to stop expecting something important to happen with CS, and move on to more fruitful balance-of-trade opportunities. However, I personally feel that CS has much to learn from IS. I see three areas of possible exchanges that move from IS to CS.

One area is "requirements". Search on the string: clear complete unambiguous requirements. People claim be able to identify clear, complete, unambiguous requirements. Some even claim to be able to teach you how to make such identifications. CS training is often predicated on the idea that it is possible to deal with requirements this way. It would certainly make system development easier. However, most IS people know this is virtually impossible. People (especially the fabled "users") have different goals, objectives, constraints, and so on, even in the same organization. People who teach in business schools have known for years that "bosses" cannot just tell "workers" what to do and how to do it. In contrast, CS people often treat goal incongruity, subversion of objectives, and conflicting understandings of issues as "management" problems that managers should fix. Getting humans out of the loop altogether might solve such problems (some engineers and computer scientists like this idea), but that is harder than it seems. This is less a problem to be solved than a situation to be embraced. Improved ways of doing "requirements" is of interest to both CS and IS, and the people from IS should have much to contribute here.

Another area lies in the design of large-scale socio-technical systems. Many of the largest computing-based activities lie in the realm of commercial computing that IS people know well: transaction processing, order entry for electronic commerce, financial clearing, and so on. With the advent of large-scale sensor networks, it will be possible to link enormous amounts of data together for analysis, a notion hinted at by the "big data" discourse (Brynjolfsson & McAfeee, 2011). Some argue that the availability of large-scale data can fundamentally change the way enterprise works, and the challenges of building such infrastructure are intriguing enough to attract both IS and CS people. Such systems require understanding of technical problems that have traditionally attracted CS, but they also entail issues of organizations and institutions that have been a singular focus of IS. Moreover, big data cannot easily be separated from big computing, big storage, and implementation trouble. Just because it is worth the trouble does not make the trouble go away. There is plenty of work to be done to keep people from CS and IS occupied. Programmatically, one of the more interesting developments that has captured people from both CS and IS the focus sparked by Simon's "science of design" (Simon, 1969). This promises to be interesting and important, and should provide IS people with many opportunities to influence CS and other fields.

Finally, there is the phenomenon of "social computing" that crosses boundaries without care for academic sensibilities. The Internet and the web took off primarily as a communication tools, which suggests that the biggest changes are in relationships among people rather than only among machines or machines and people. Now we are seeing that phenomenon again as social computing grows at an amazing pace. Social computing makes academic borders such as those between CS and IS moot. Will CS and IS people relax their vigilance and go with the phenomena at hand? If they do not, the people behind them will: as Max Planck suggests, science advances funeral by funeral (Planck, 1949, p. 33-34). CS and IS will probably encounter considerable pressure from students who do not care much about such borders, especially if the work they want to do ignores the borders altogether. IS should have much to contribute to CS in this endeavor.

IS should have more significant influence on CS as new areas of IT application arise. But to achieve such influence, the IS field must move beyond ways of thinking that have long dominated management education. The revolution described in the PCAST-NITRD report is echoed in the rise of social networking technology, the implications of massive sensor networks, the growth in computationally-enabled control systems, cloud computing, and many other areas that have the potential to fundamentally alter human enterprise. Traditional management education, which remains grounded in the 20th century orthodoxy of industrial administration, cannot easily embrace these new forces (King, 2011). The future of IS research has never been brighter, but the spark to light it must come from IS researchers who are willing to look beyond the constraints of the past.

Acknowledgments

The author thanks Cynthia Beath, Nick Berente, Jonathan Grudin, Kalle Lyytinen, and Susan Winter for comments on this paper.

References

Bass, J. (1992, November). Controversy over NRC report resolved; NRC, petitioners issue joint statement. *Computing Research News*, *4*(5), 1-12. Retrieved from http://archive.cra.org/CRN/issues/9205.pdf

Boland, R. J. Jr., Singh, J., Silipante, P., Aram, J. D., Fay, S. Y., & Kanawattanachai, P. (2001). Knowledge Representation and Knowledge Transfer. *The Academy of Management Journal*, *44*(2), 393-417

Brynjolfsson, E., & McAfee, A. (2011, November). The big data boom is the innovation story of our time. *The Atlantic.* Retrieved from http://www.theatlantic.com/business/archive/2011/11/the-big-data-boom-is-the-innovation-story-of-our-time/248215/

Hartmanis, J., & Lin, H. (Eds.). (1992). *Computing the future: A broader agenda for computer science and engineering.* Washington, DC: National Academies Press.

King, J. L. (2010). Project SAGE, a half-century on. *Interactions*, *5*, 53-55.

King, J. L. (2011). CIO – concept is over. *Journal of Information Technology*, *26*, 129-138.

King, J. L., & Lyytinen, K. (Eds.). (2006). *Information systems: The state of the field*. West Sussex, UK: John Wiley and Sons.

Lyytinen, K., & King, J. L. (2004). Nothing at the center? Academic legitimacy and the information systems field. *Journal of the Association for Information Systems, 5*(6), 220-246.

Orlikowski, W. J., & Iacono, C. S. (2001). Research commentary: Desperately seeking the “IT” in IT research - a call to theorizing the IT artifact. *Information Systems Research*, *12*(2), 121-134.

PCAST. (2010). Designing a digital future: Federally funded research and development in networking and information technology. *Office of the President of the United States.* Retrieved from http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-nitrd-report-2010.pdf

Planck, M. (1949). *Scientific autobiography and other papers* (Frank Gaynor, trans.). New York: The Philosophical Library.

Sawyer, S., & Winter, S. (2011). Special issue on futures for research on information systems: Prometheus unbound? *Journal of Information Technology*, *26*, 95-98.

Simon, H. (1969). *The sciences of the artificial*. Cambridge, Mass: MIT Press.

Winter, S. J., & Butler, B. S. (2011). Creating bigger problems: Grand challenges as boundary objects and the legitimacy of the information systems field. *Journal of Information Technology*, *26*, 99-108.

Yates, J. (2005) *Structuring the information age: Life insurance and technology in the twentieth century*. Baltimore: Johns Hopkins University Press.

About The Author

John Leslie KING is W.W. Bishop Professor of Information and former Dean of the School of Information and former Vice Provost at the University of Michigan. He joined Michigan in 2000 from the faculties of computer science and management at the University of California at Irvine. He was Editor-in-Chief of *Information Systems Research* and has been associate editor of many other journals. He has been a member of the Board of the Computing Research Association (CRA) and the Council of the Computing Community Consortium, as well as Advisory Committees for the National Science Foundation's Directorates for Computer and Information Science and Engineering (CISE) and Social, Behavioral and Economic Sciences (SBE), and NSF's Advisory Committee for Cyberinfrastructure (ACCI). He holds a PhD in administration from the University of California, Irvine, and an honorary doctorate in economics from Copenhagen Business School. He is a Fellow of the Association for Information Systems and a Fellow of the American Association for the Advancement of Science.